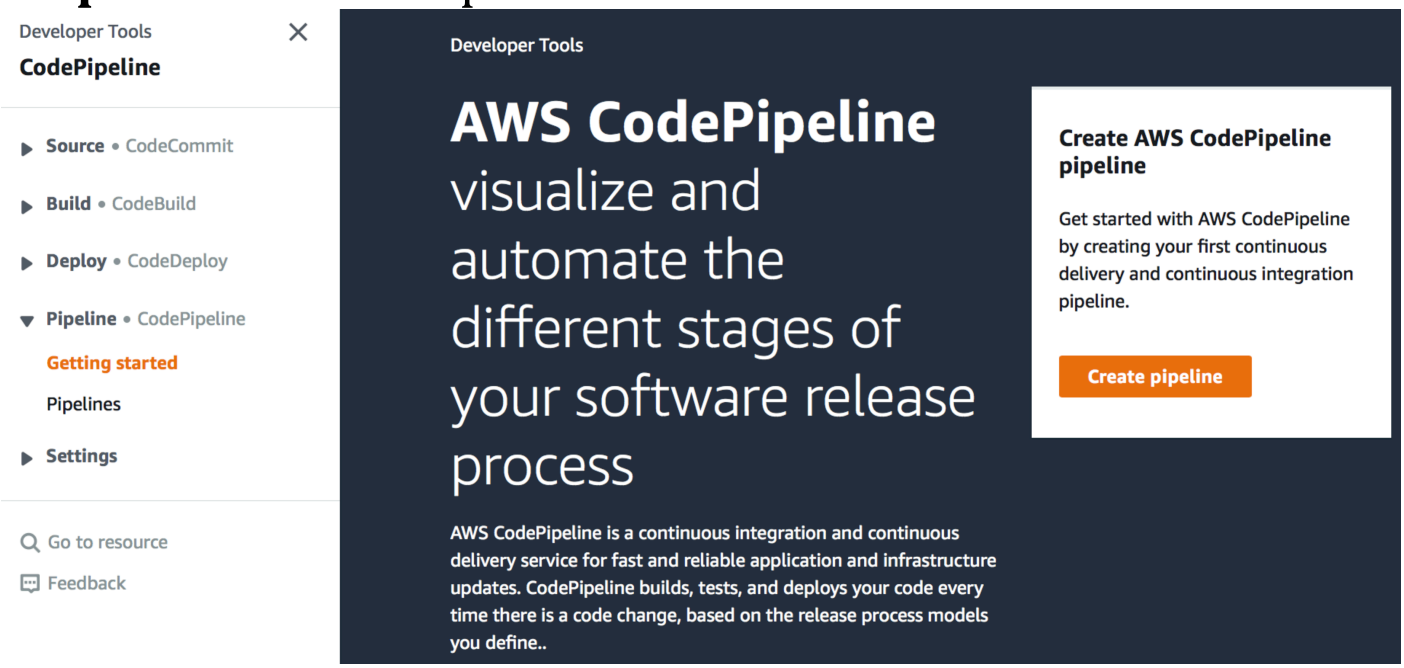
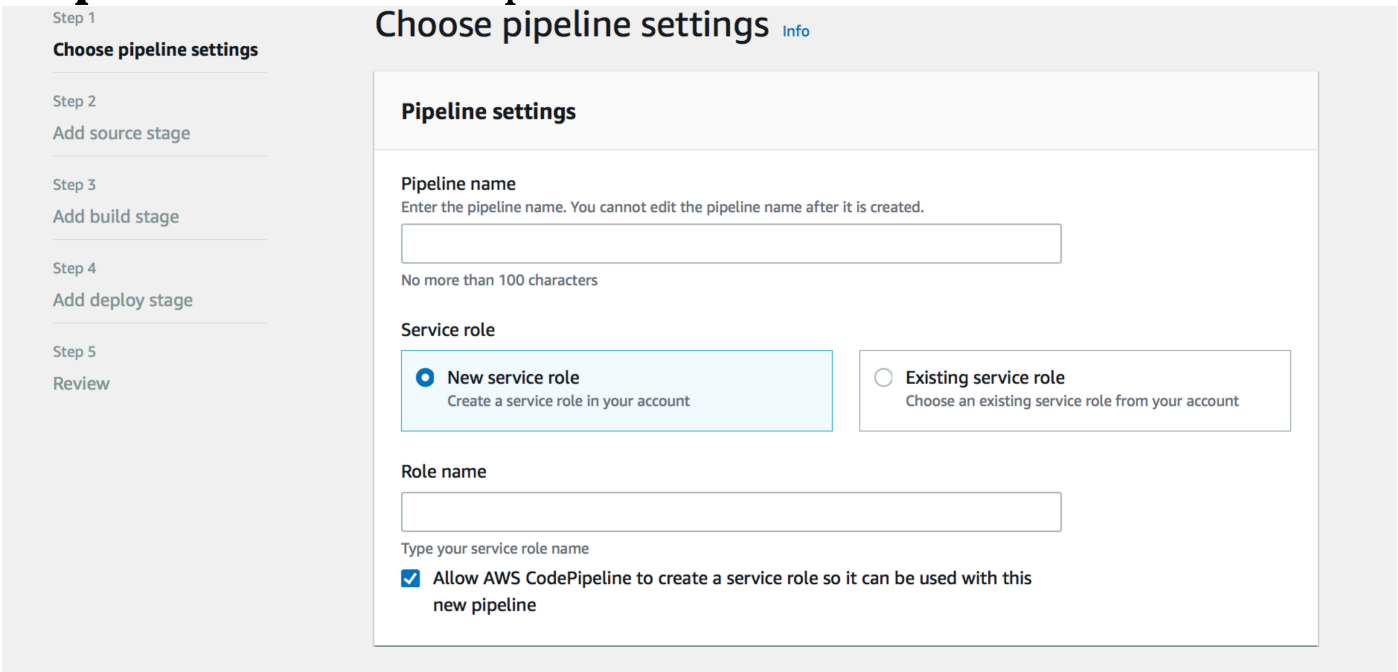
Exercise 06.1:CodePipeline :Deploying a sample webapp using CodePipeline

Step 1: Start your EC2 instances with tags Dev and Production

Step 2: Click CodePipeline



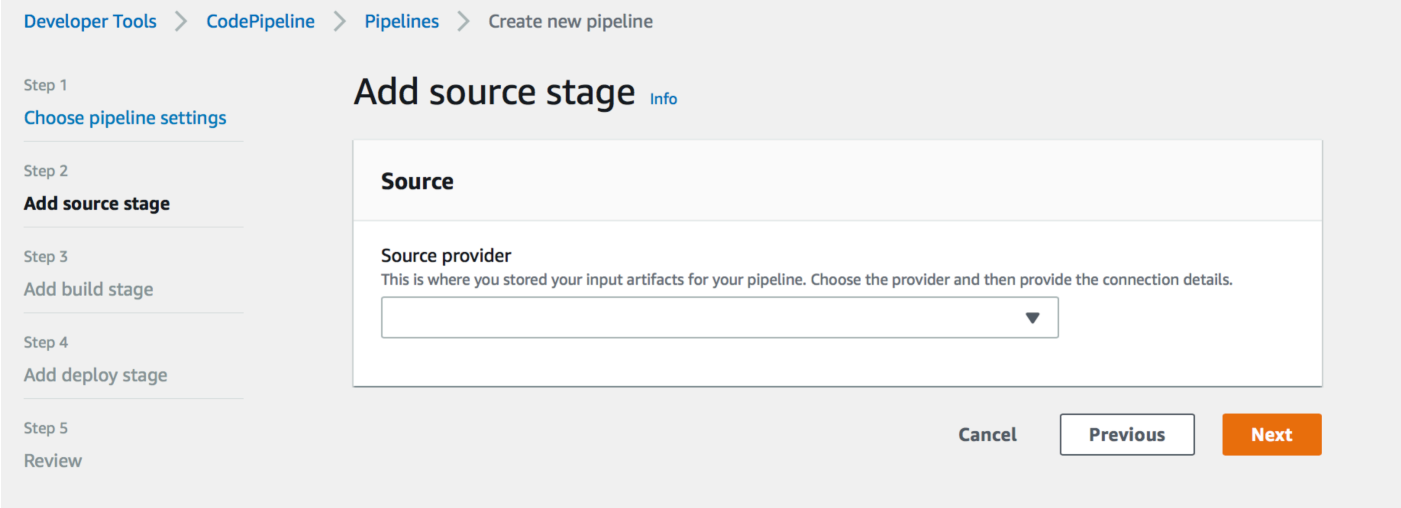
Step 3: Click Create Pipeline



Step 4: Give the following details
Pipeline name: my-webpage-pipeline
Service role: Default
Advanced Settings:

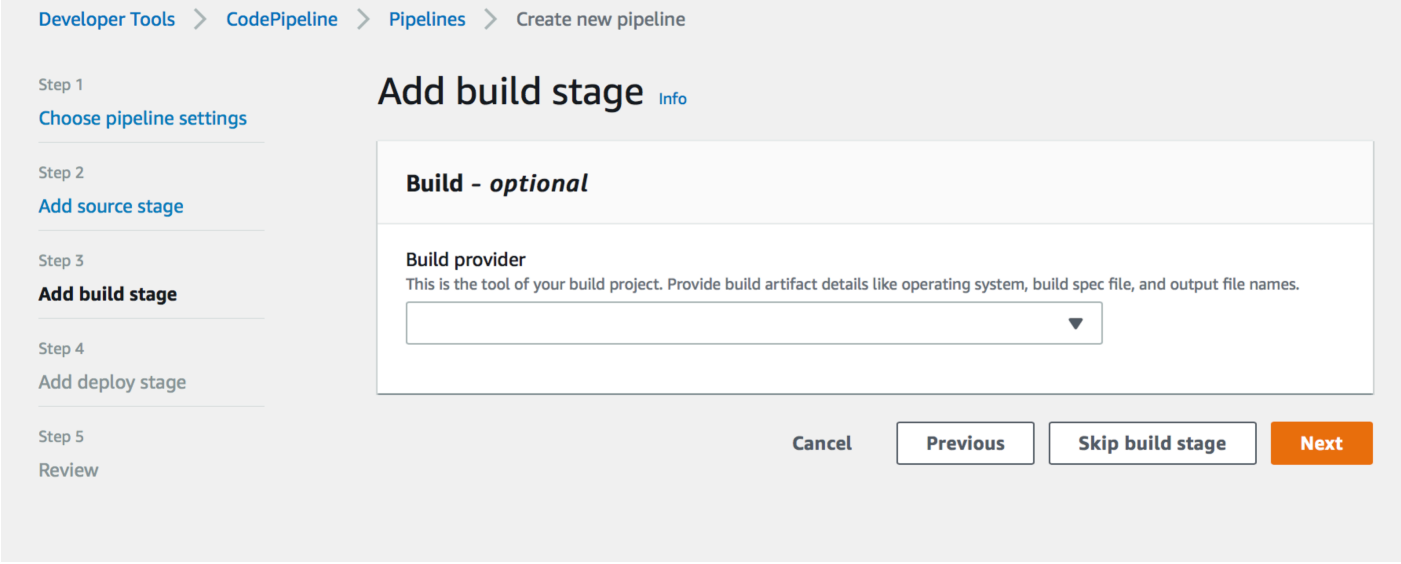
- **Custom Location:** Create a bucket and choose it.
- **Encryption Key:** Default

Click Next

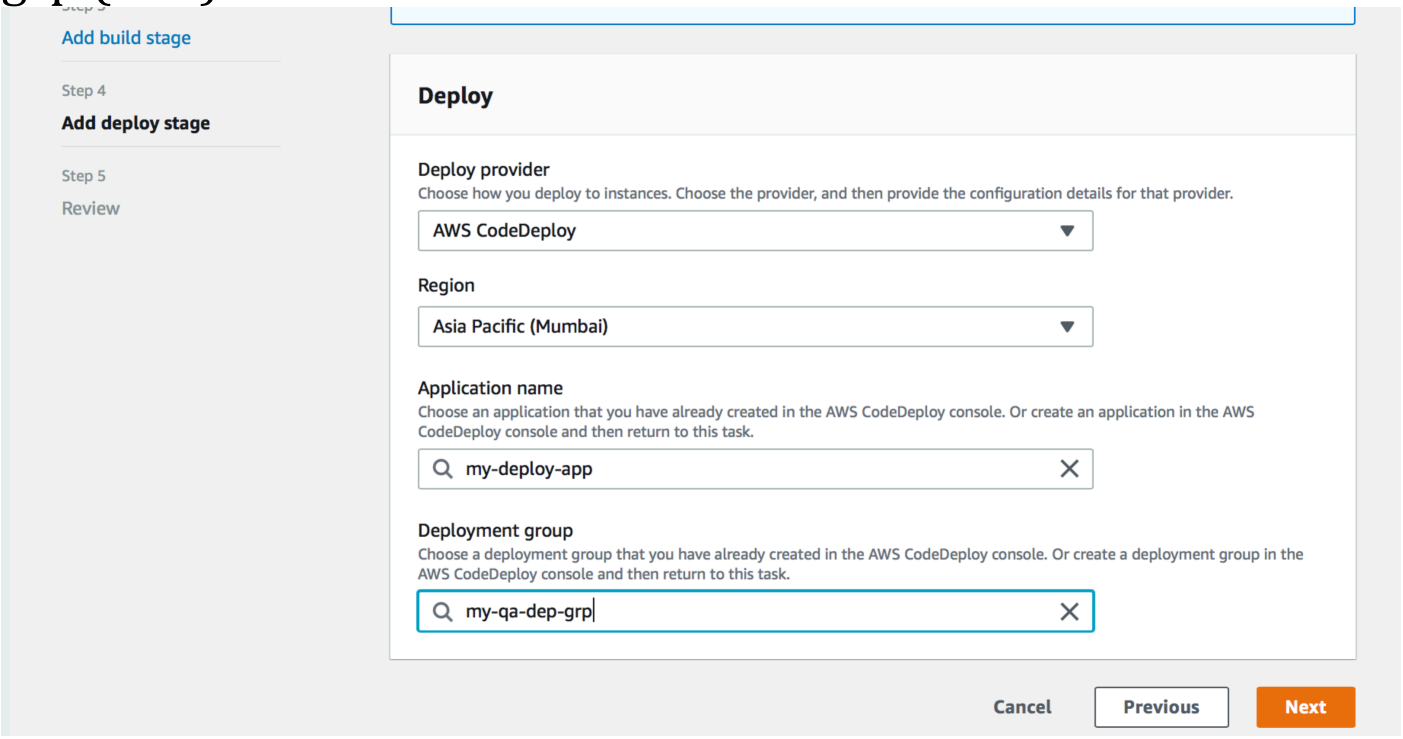


Step 5: Select CodeCommit, Repository name and Branch , Change DetectionOptions=Default click Next

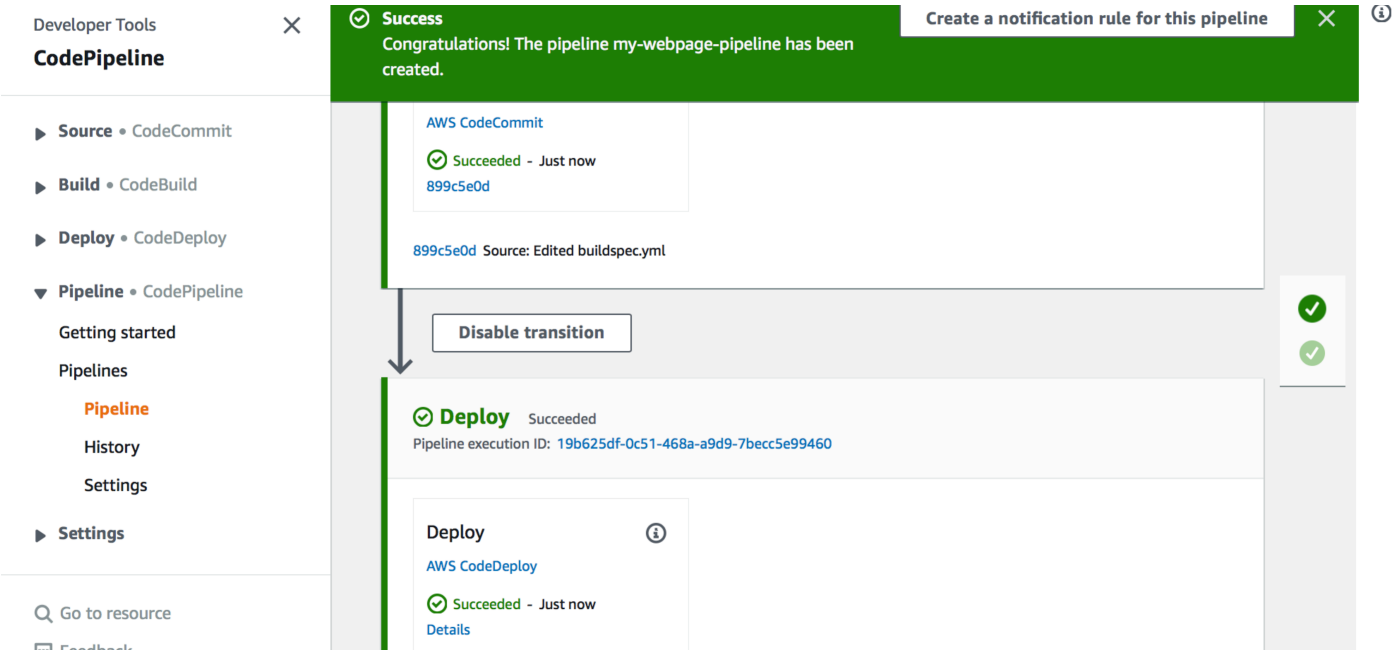
Step 6: Click Skip build stage.



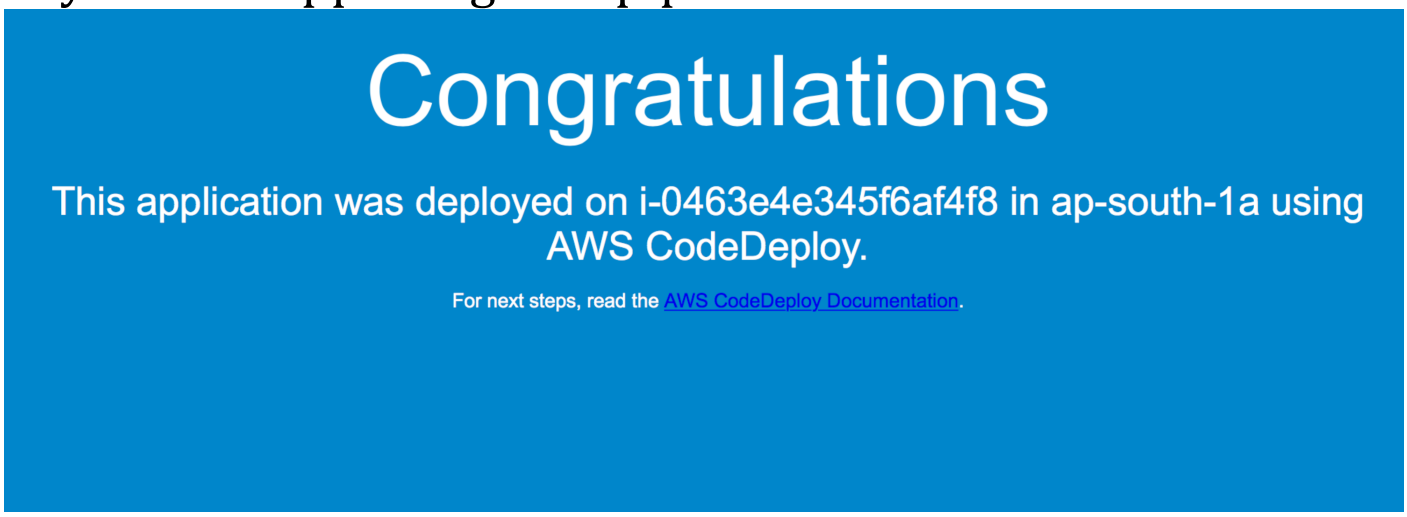
Step 7: Select Code Deploy,Region ,App name and deployment grp (Dev) and click Next



Step 8: Review the details and click Create Pipeline. You should see a successful completion screen as follows if everything went successful



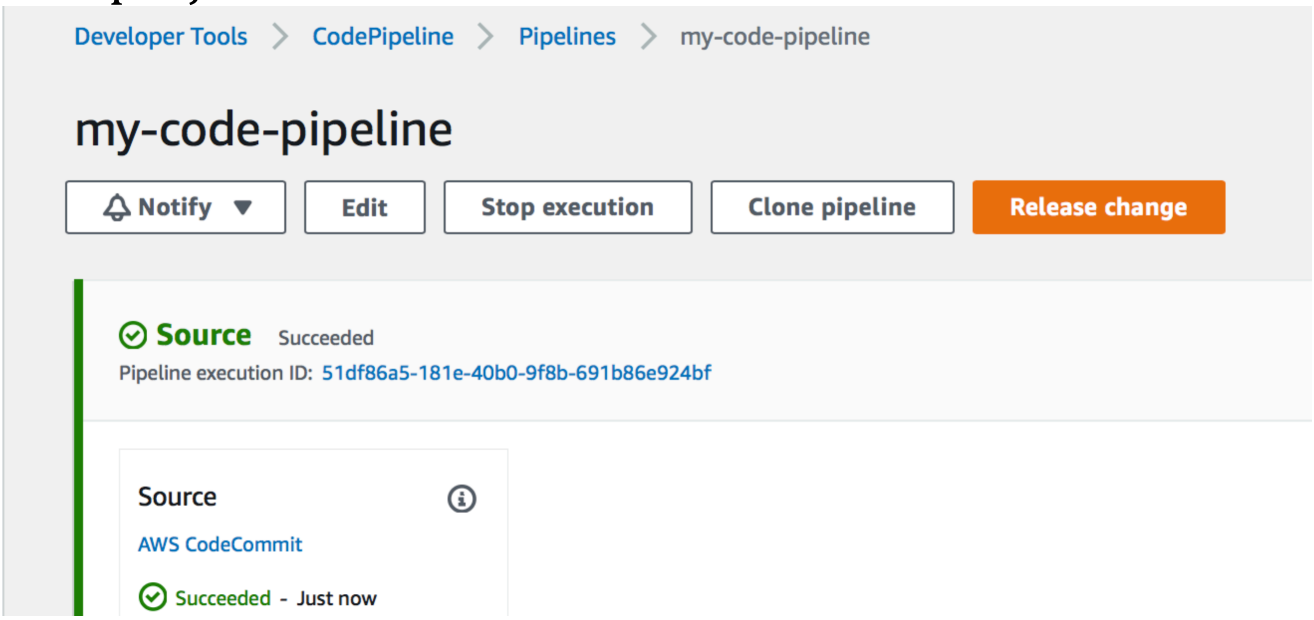
Step 9: Go to the browser and type the Public IP to get a congratulations page which confirms the successful deployment of your webapp using codepipeline



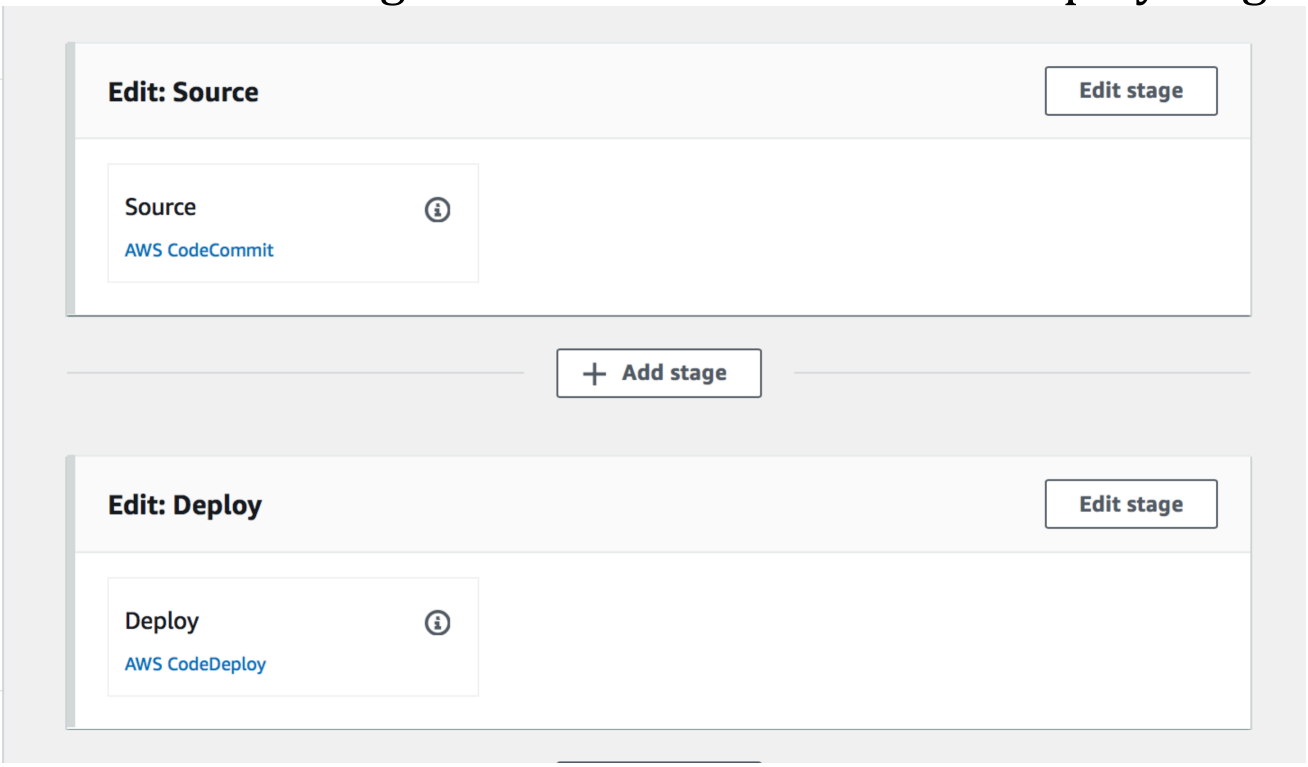
Step 10: Make some changes in the index.html and commit the code. The changes should be readily available on the webpage as codepipeline initiated another automatic deployment

Adding CodeBuild to the existing pipeline

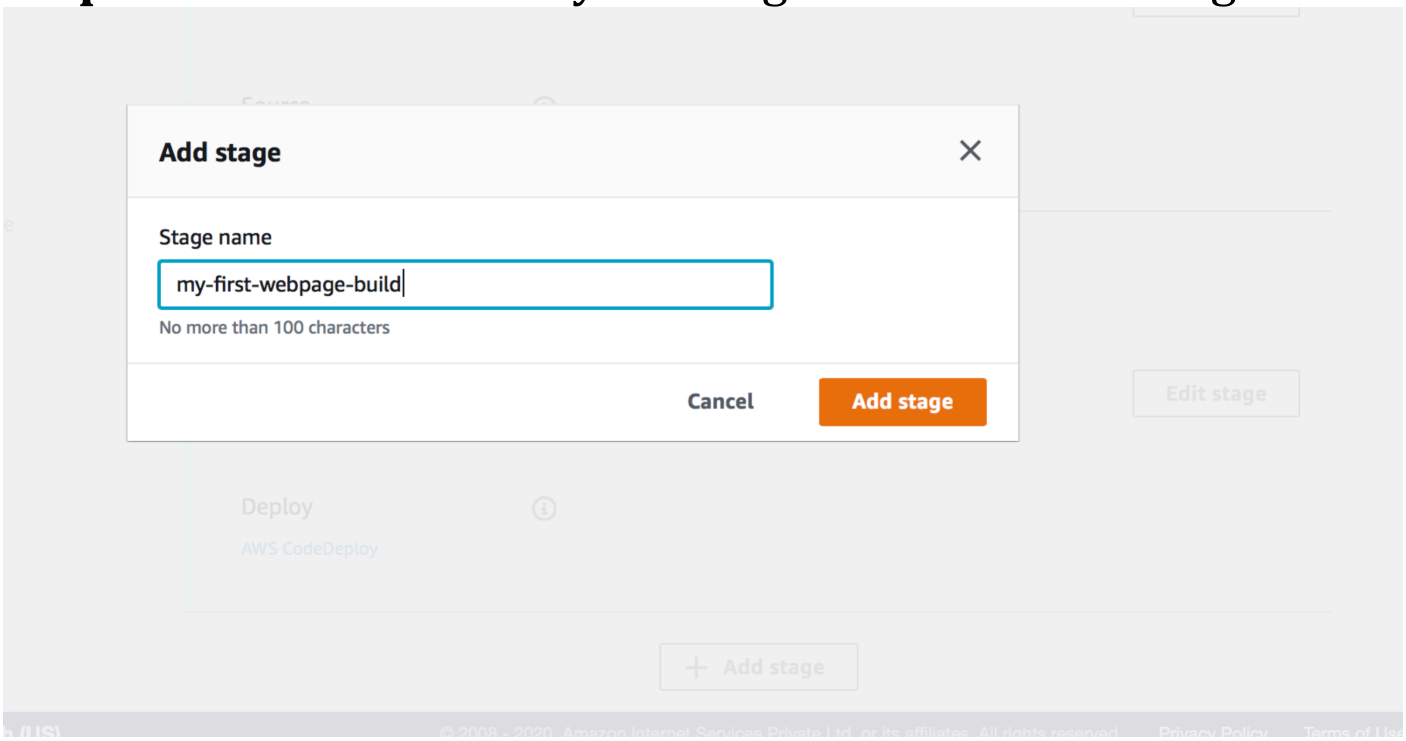
Step 1: Edit the existing pipeline by clicking the edit button in the pipeline project



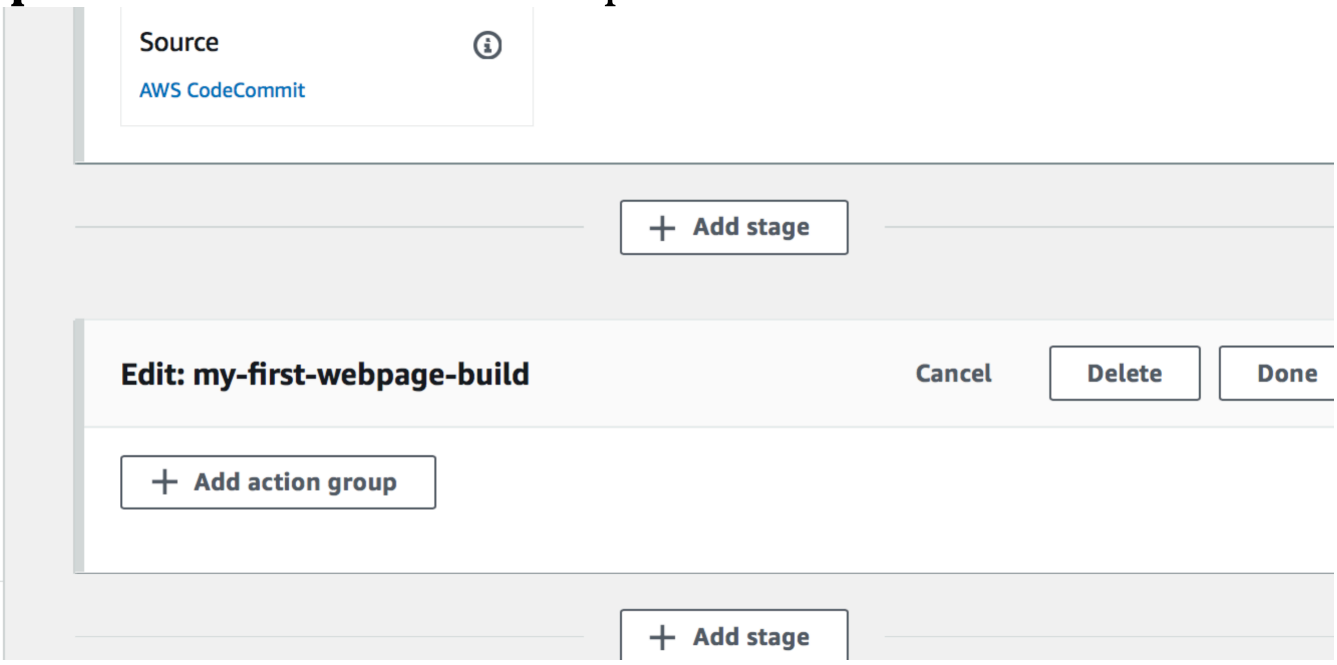
Step 2: Click Add Stage between the Source and Deploy stage.



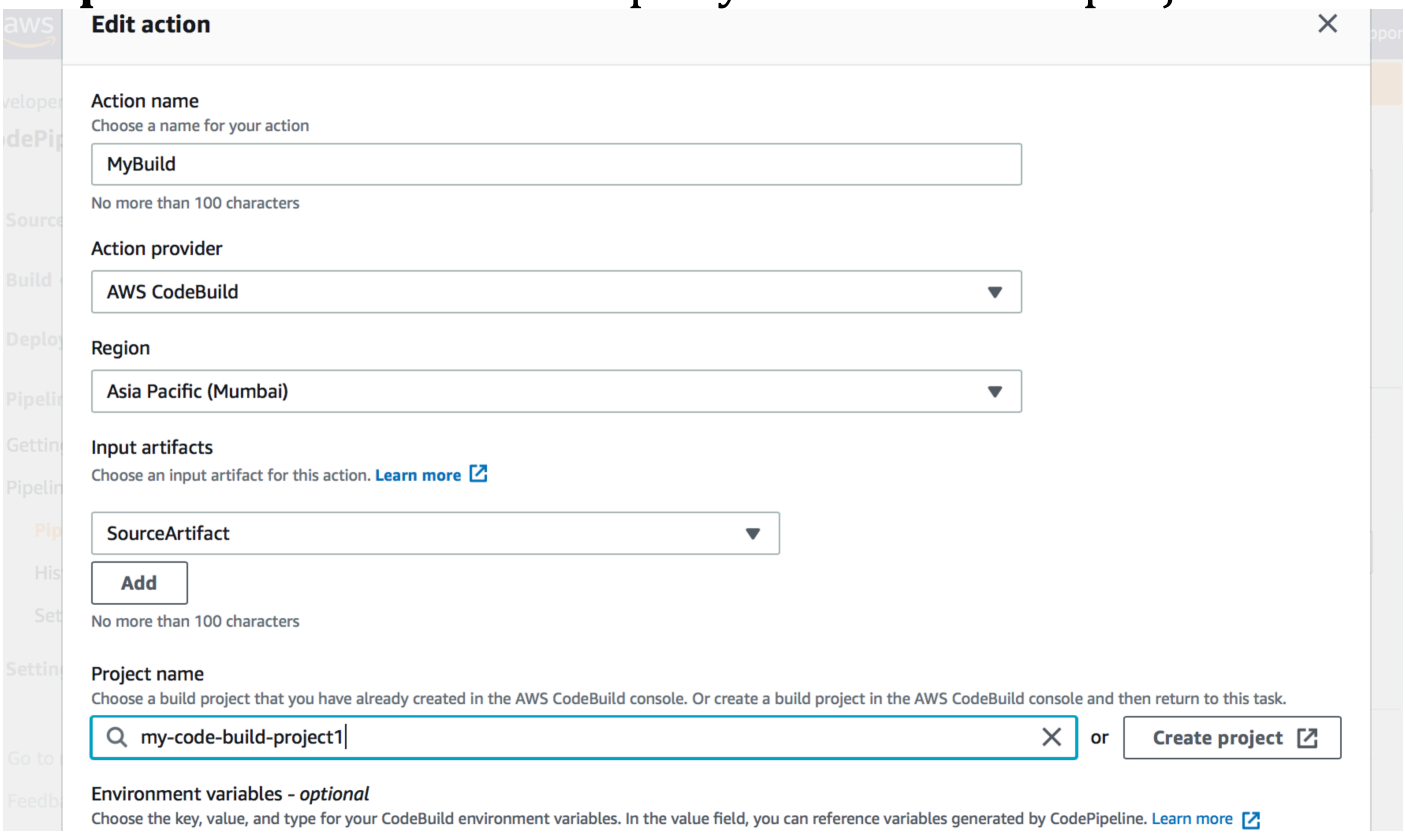
Step 3: Enter a name for your Stage and click Add stage



Step 4: Click Add action Group



Step 5: Enter the details as per your CodeBuild project

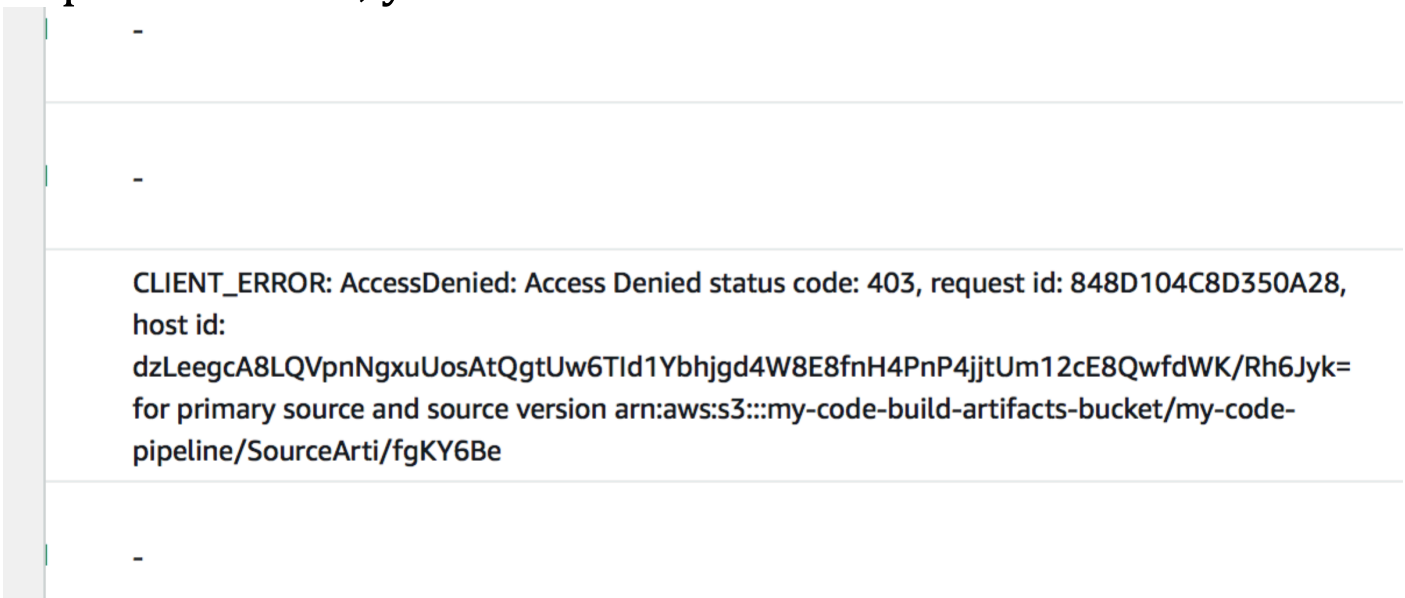


Step 6: Click Save

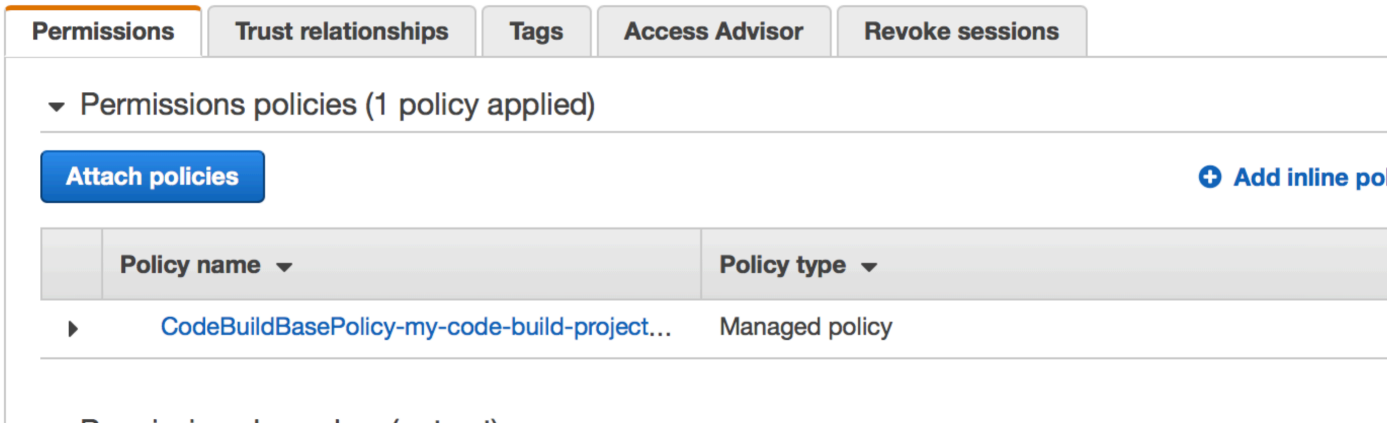
Step 6.1: Edit the Deploy Stage to have the source as TestOutput

Step 7: Click Release Change. This should start the build.

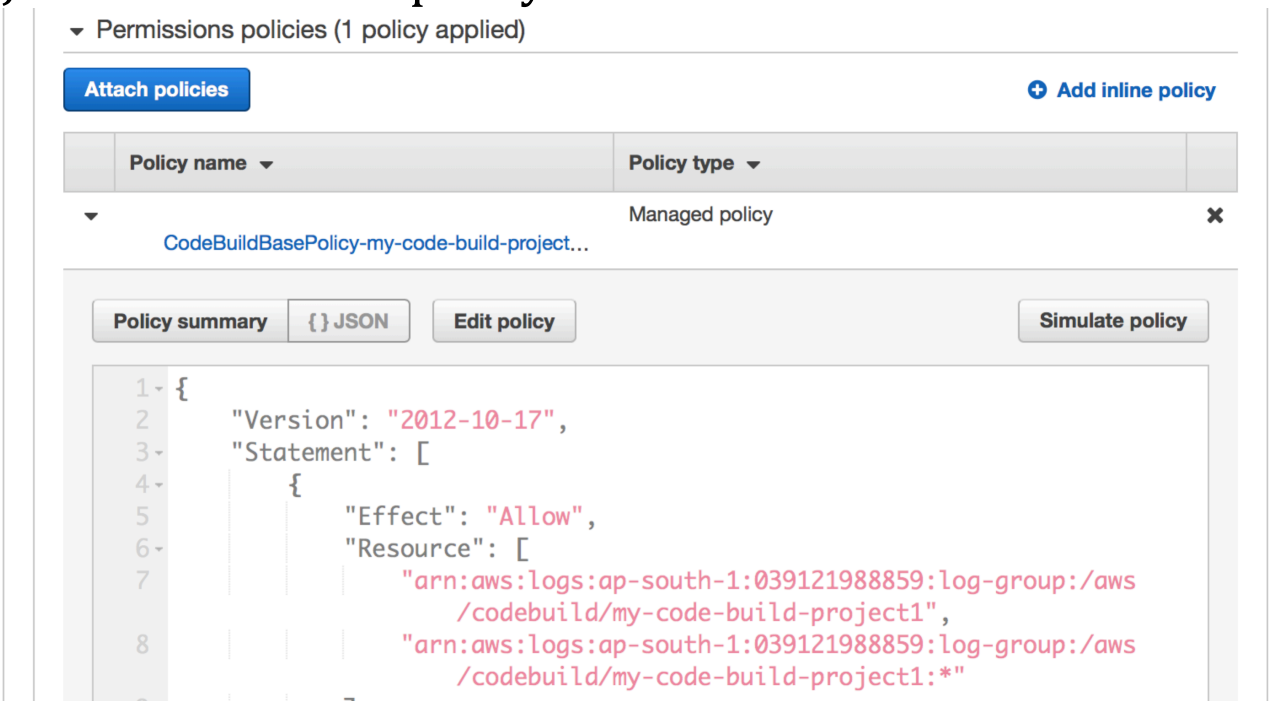
Step 8: If you get failed status click details under the error and go to phase details, you can see the error as shown below.



Step 9: Got to IAM→ roles→ search for codebuild and expand the codebuild rule

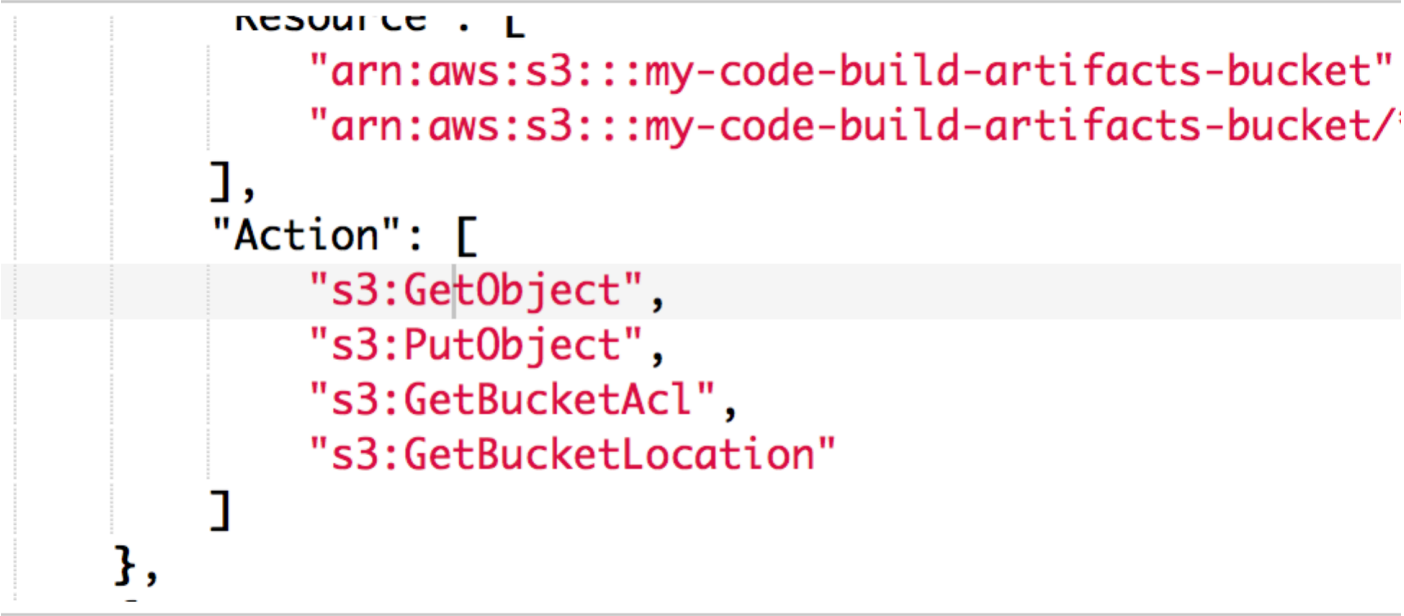


Click JSON tab and edit policy



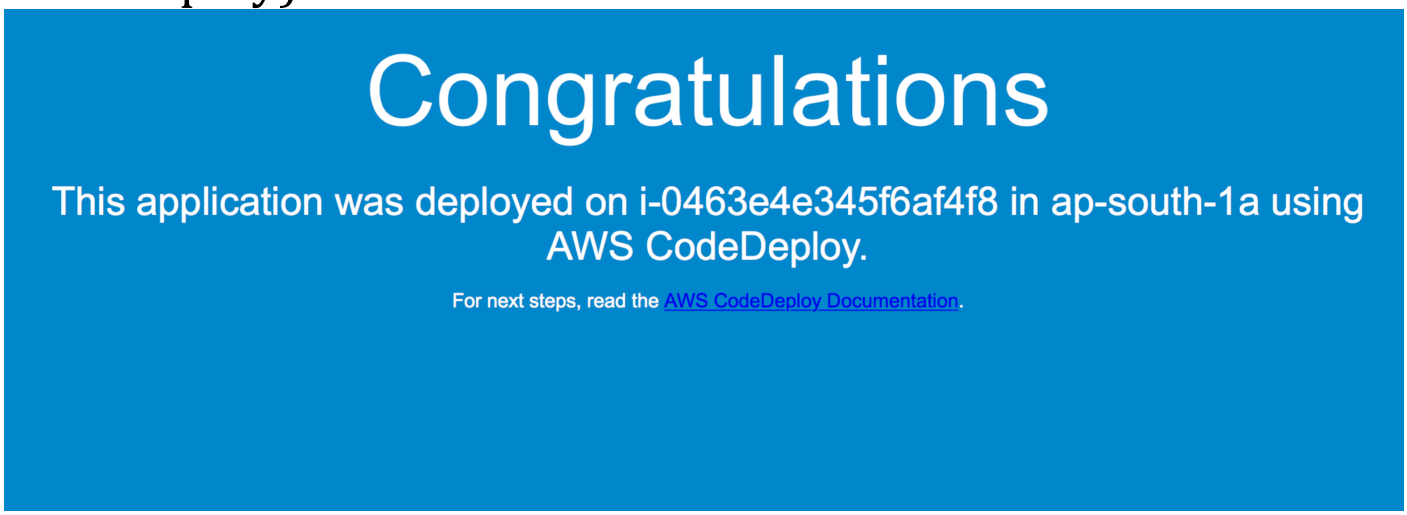
Go to the rule where you have your bucket

Add a new Action “s3:GetObject” at the top as below and click Review and save .



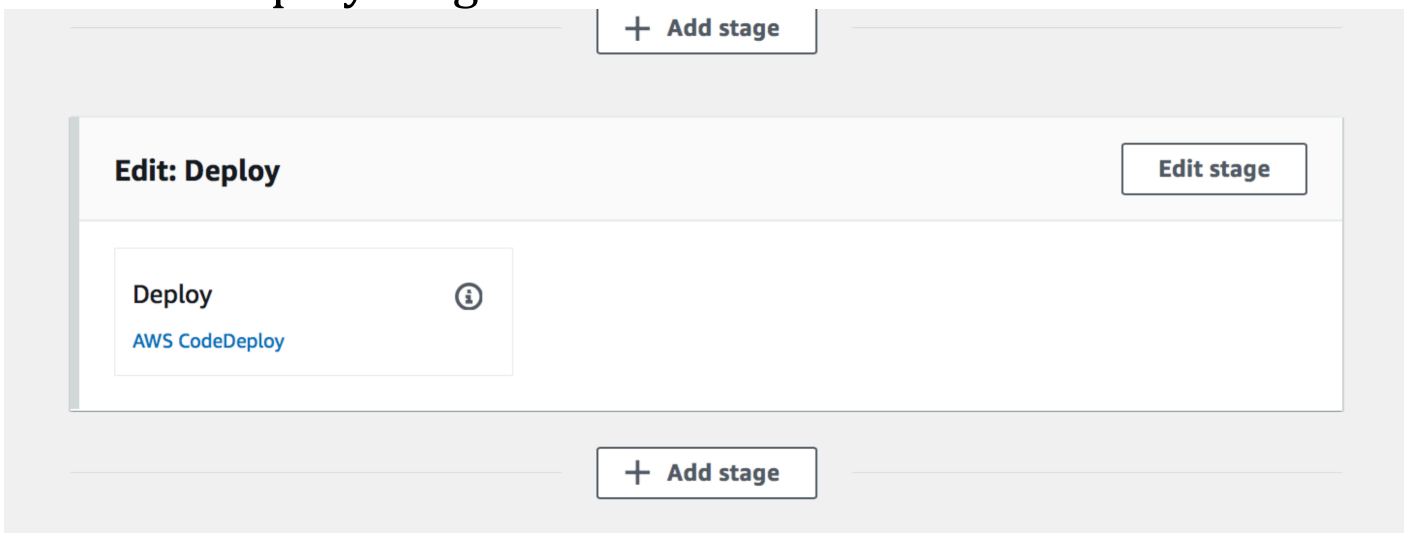
Step 10: Go back to the pipeline and click retry on the build stage.
Step 11: You should see successful on all the phases.
Step 12: Go to the browser and type the Public IP to get a congratulations page which confirms the successful deployment

of your webapp using codepipeline (CodeCommit, CodeBuild, CodeDeploy)

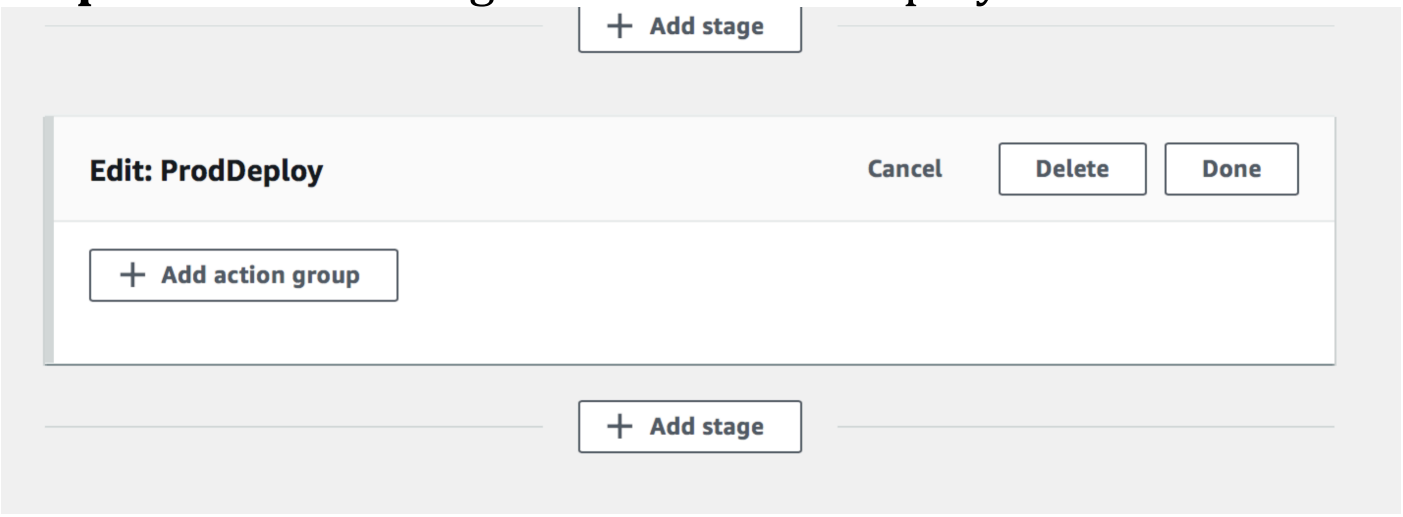


Adding a new Production stage and adding a manual approval process

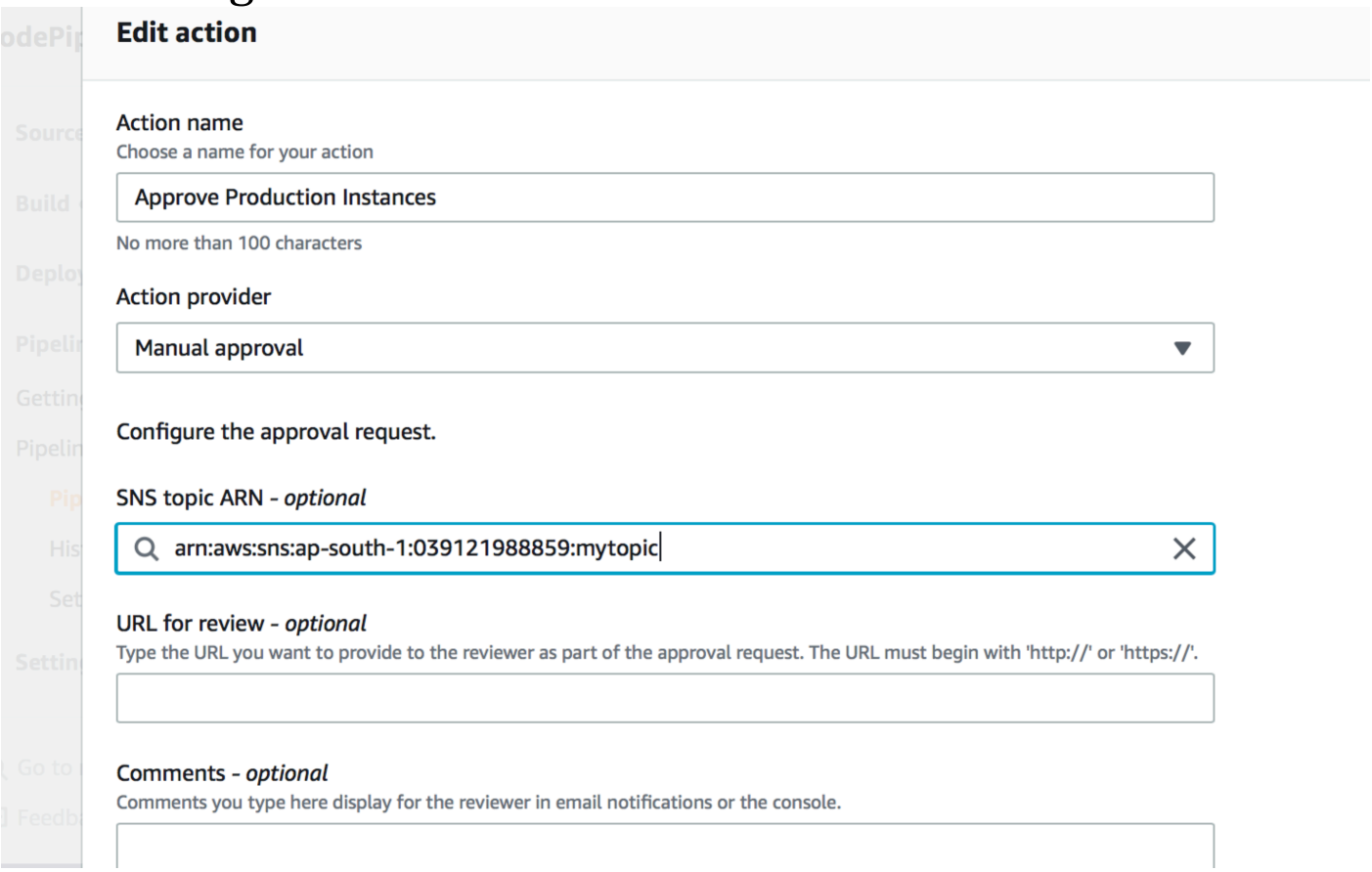
Step 1: Edit the existing code pipeline project and click Add Stage after the Deploy Stage



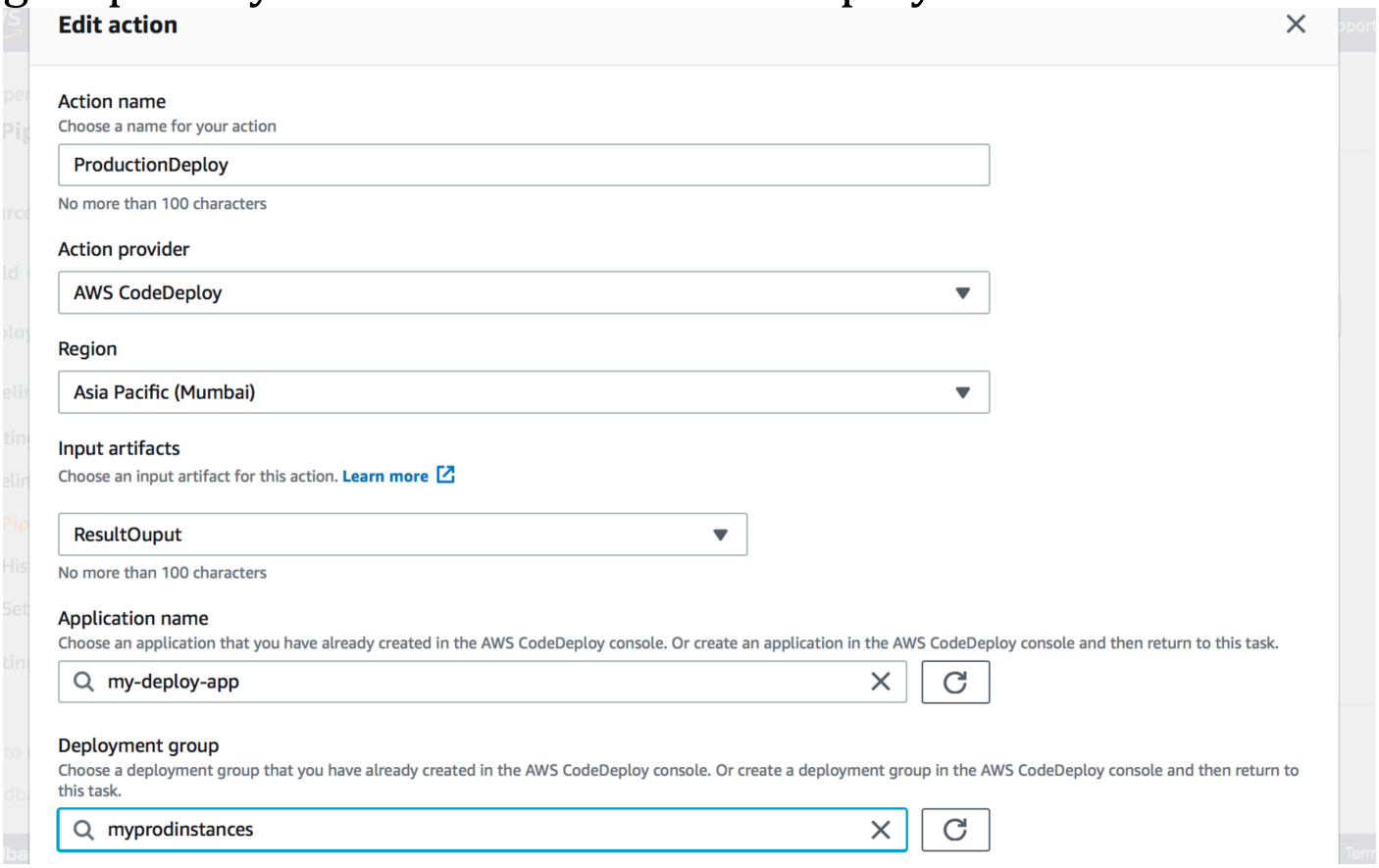
Step 2: Enter the stage name as ProdDeploy and click Add Stage



Step 3: Click Add Sction Group and give an action name, Action Provider as “Manual Approval” and optianally a SNS topic if you have configured

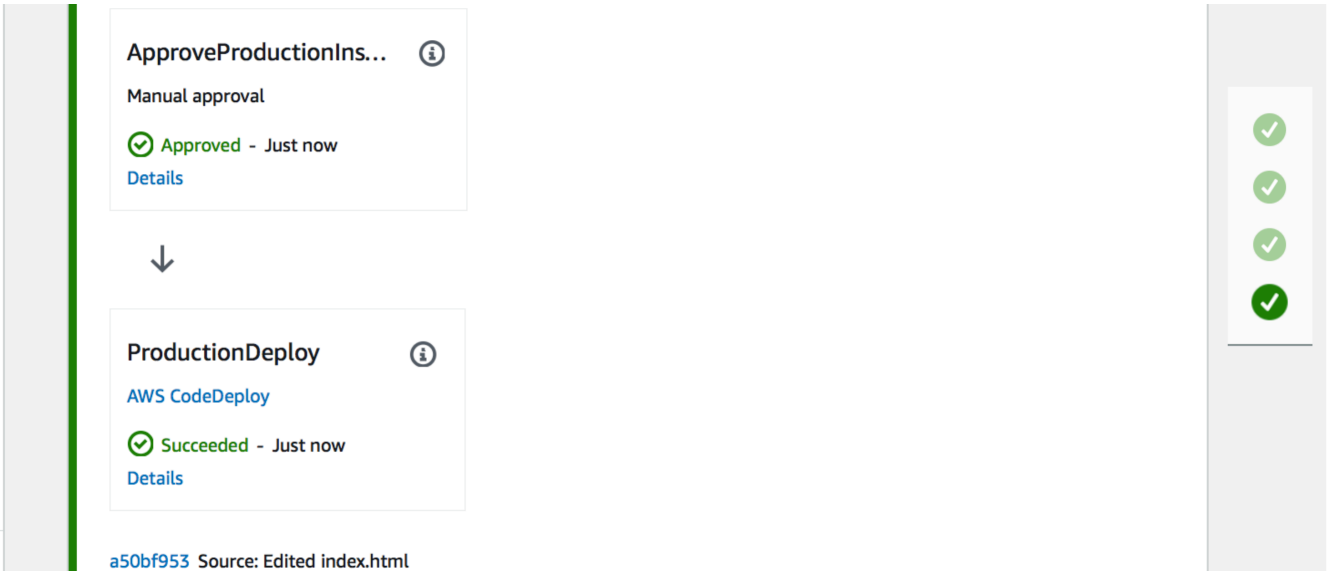


Step 4: Add another action group below it, for production deployment. Select an action name, Action provider=CodeDeploy, InputArtifacts=ResultOutput(from code build stage), deploy application,and deployment = Your production deployment group that you should have in codedeploy module



Step 5: Click Done and Save and then click Release change to start the code pipeline. The pipeline execution should stop at the approval stage for review

Step 6: Click the Review button and provide approval comments to kick start the deplyment to production instances.



Step 7: Test your application by pointing the production instance IP in the web page